UNITED STATES PATENT APPLICATION

FOR

BI-DIRECTIONAL TEXT SUPPORT IN LEGACY APPLICATIONS

Inventor:
Bruce V. Schwartz

Prepared by:
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025-1026
(408) 720-8598

Attorney's Docket No: 03399.P026

"Express Mail" mailing label number: _EL 6390151833US_
Date of Deposit: _April 25, 2000_
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231
Carrie Boccaccini
(Typed or printed name of person mailing paper or fee)
Carrie Boccaccini
(Signature of person mailing paper or fee)
_4-25-2000_
(Date signed)

BI-DIRECTIONAL TEXT SUPPORT IN LEGACY APPLICATIONS

BACKGROUND OF THE INVENTION

Field of the Invention

5   The invention relates to user interfaces and more particularly relates to display of text from legacy applications written for unidirectional text in bi-directional text environments.

Description of the Related Art

10   Handling bi-directional text presents challenges to designers of user interfaces. In one example, the bi-directional text may include both native text which is read right-to-left and foreign text which is read left-to-right. Thus, the bi-directional text cannot be simply printed or displayed one character after another. Unidirectional text, in contrast, is text which is all left-to-right (or right-to-left) text. With bi-directional text, native text, such as Hebrew or Arabic, should be displayed in a right-to-left order, while the foreign text, such as English, should be displayed in a left-to-right order. An example would be an article reported in Arabic or Hebrew that contains quotes or terms of art in English that are not translated. In such an article, the main text of the article, the Arabic or Hebrew text, is read right-to-left, but the untranslated English portions are read left-to-

20   right. While this is conceptually simple for the reader, laying out this text may prove difficult.

In particular, if the text is stored in the order in which it is to be displayed or printed, then manipulation of the text may prove complex because of the embedded

- 1 -

portions of the text which are oriented backwards relative to the rest of the text. Moreover, on mobile computing devices, bi-directional text may be important when the device is in an area where such text is utilized, but unidirectional text may be important when the device is in an area where text only flows in one direction.

5   Typically, software is written to either handle right-to-left or left-to-right style text, not both. Moreover, most software is developed in locations where people read left-to-right, so compatibility for right-to-left or bi-directional text is not considered. Typically, text is stored as a string of characters in memory as the characters would be printed in a left-to-right fashion. Thus, it is potentially valuable to create software which may be

10   used to adapt software written solely for left-to-right text for use in a bi-directional text environment. A bi-directional text environment may be defined as either an area where bi-directional text is utilized, or a device in which support for bi-directional text is part of the functionality of the device or its software.

SUMMARY OF THE INVENTION

A method and apparatus for support of bi-directional text in legacy applications is disclosed. The method may be used in different embodiments for either monospace or proportionally spaced fonts. In general, the method involves first flipping all of the text, and thereby putting most of the text into its proper position and orientation. The method then involves finding any foreign (left-to-right) text within the flipped text, and flipping it back.

5

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the accompanying figures.

Figure 1A illustrates unadjusted bi-directional text.

Figure 1B illustrates adjusted bi-directional text.

Figure 2A illustrates unadjusted bi-directional text.

Figure 2B illustrates adjusted bi-directional text.

Figure 3A illustrates unadjusted bi-directional text.

Figure 3B illustrates adjusted bi-directional text.

Figure 4A illustrates unadjusted bi-directional text.

Figure 4B illustrates partially adjusted bi-directional text.

Figure 4C illustrates adjusted bi-directional text.

Figure 4D illustrates flipping of characters within a line of text.

Figure 4E illustrates flipping of characters within a run of text.

Figure 5A illustrates unadjusted bi-directional text in a proportional font.

Figure 5B illustrates partially adjusted bi-directional text in a proportional font.

Figure 5C illustrates adjusted bi-directional text in a proportional font.

Figure 6 illustrates an embodiment of a system.

Figure 7 illustrates an embodiment of a method of adjusting text in a monospace font.

Figure 8 illustrates an embodiment of a method of adjusting text in a proportional font.

Figure 9 illustrates an exemplary medium embodying a method of adjusting text.

Figure 10 illustrates an embodiment of a general purpose computer.

Figure 11 illustrates an embodiment of a network including servers as connected

to a wireless device.

Figure 12 illustrates an embodiment of a wireless device which may be

5   connected to a network.

DETAILED DESCRIPTION

A method and apparatus for bi-directional text support in legacy applications is described. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the invention. It will

5   be apparent, however, to one skilled in the art that the invention can be practiced without these specific details. In other instances, structures and devices are shown in block diagram form in order to avoid obscuring the invention.

Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic described in connection with the

10   embodiment is included in at least one embodiment of the invention. The appearances of the phrase "in one embodiment" in various places in the specification are not necessarily all referring to the same embodiment, nor are separate or alternative embodiments mutually exclusive of other embodiments.

The method, in one embodiment, may be used in different embodiments to

15   support display or printing of bi-directional text for either monospace or proportionally spaced fonts. Monospace fonts are fonts in which all letters (or all characters) are of the same width, and are generally found on simple displays. Proportionally spaced fonts are fonts having different widths for each letter in the font. In general, the method involves first flipping all of the text of a line of text about the center vertical axis of the

20   line (including any leading or trailing blank space), and thereby putting most of the text into its proper position and orientation. The method then involves finding any foreign (left-to-right) text within the flipped text, and flipping it back about the center of the foreign text within the area where the foreign text is located, thus putting that text into its

proper orientation. As a result, the text is in a condition such that it may be displayed in the order and format in which it is stored.

The method, in one embodiment, may be particularly suitable for mobile computing devices of various complexity. Such devices typically have small screens for

5   which the storage (memory) for the entire display is not large. As a result, the brute-force approach of flipping all text and then flipping back foreign text may not be computationally expensive. Moreover, the lack of need for data structures for holding text to be flipped can further improve performance in such devices.

Figure 1A, 1B, 2A, 2B, 3A and 3B all illustrate a before-and-after perspective of

10   text, before and after the method is performed on the text. Each figure depicts three lines of text on a small display. In each figure, capitalized letters represent text which should be read (and therefore displayed) right-to-left. Lowercase letters represent text which should be read and displayed left-to-right. Thus, in Figure 1A, all of the text ("PLEASE CHECK OUR WEB SITE FOR UP DATES") is capitalized (representing text

15   in a language such as Hebrew or Arabic for example) and should therefore be displayed right-to-left. In Figure 1B, the text of Figure 1A is illustrated in right-to-left orientation (after the method is performed).

Similarly, most of the text ("PLEASE CHECK WITH" and "FOR UPDATES.") in Figure 2A is capitalized (right-to-left), with the exception of the text 'joe clemens' which

20   is lowercase (left-to-right). Figure 2B illustrates the resulting display of the text after the method is performed. Likewise, all of the text ("PLEASE CHECK" and "FOR UPDATES.") of Figure 3A is right-to-left, except for the two portions, 'joe' and 'clemens' which are left-to-right. The resulting display of the text after the method is performed is

illustrated in Figure 3B. In each of these figures, one of the three general cases of bi-directional text is illustrated. In Figures 1A and 1B, no foreign text is present. In Figures 2A and 2B, foreign text is present on one line. In Figures 3A and 3B, foreign text is present on multiple lines. Note that the punctuation follows the characters it is

5   associated with.

The two portions 'joe' and 'clemens' may be referred to as runs. A run is a group of characters which are adjacent to each other for purposes of display and which share a common set of characteristics. For purposes of this discussion, runs do not span lines. In general, a run may be defined as a set of characters which share the same

10   font, the same style, the same type-size, or other characteristics. However, for this discussion, we will refer to runs of characters which are either all right-to-left characters or all left-to-right characters.

Turning to Figures 4A, 4B and 4C, the transformations the text undergoes for monospace fonts are illustrated. For each of these figures, characters with an H

15   represent characters in Hebrew or Arabic, which should be displayed right-to-left. Characters with an F represent foreign (non-Hebrew or non-Arabic as appropriate) characters, which should be displayed left-to-right. The text as illustrated in Figure 4A is in the form in which it would be stored in memory, in the order in which it would be displayed or printed in left-to-right fashion. In Figure 4B, the text is illustrated after it

20   has been broken down into lines and reversed or flipped about the center vertical axis of the line within each line. Note that no change has been made to the orientation of each individual character, rather the relative locations of the various characters have been rearranged. In Figure 4C, the characters within the runs of foreign characters

have been flipped again within the run (flipped about the vertical center axis of the run), thus resulting in a display which has right-to-left characters and left-to-right characters (a bi-directional display) properly positioned.

Figure 4D illustrates flipping characters about a vertical center axis of a line.

5 Similarly, Figure 4E illustrates flipping characters about a vertical center axis of a run. In each case, the characters are flipped or exchanged between positions on opposite ends of the ordered group of characters which are to be flipped. Note that for odd numbers of characters, such as the line for Figure 4D and the run of Figure 4E, the center character does not move, it is not exchanged.

10 Turning to Figures 5A, 5B and 5C, the transformations the text undergoes for proportional spaced fonts are illustrated. In each figure, three runs of text are illustrated, along with the vertical center axis of the display in which the three runs appear. Runs 510 and 530 contain text which should be displayed right-to-left. Run 520 contains text which should be displayed left-to-right. Each of the three runs 510, 15 520 and 530 have a first end which is marked by shading and indicates the start of the text in that run. In Figure 5A, all three runs are displayed as they are stored in memory. Note that each run is a box of text with a position, as characters in proportional spaced fonts may not simply be specified by which character location on the screen should be occupied.

20 Turning to Figure 5B, each of the three runs is flipped in its position about the vertical center axis of the display. Note that this not only means that the run moves from one side of the screen to the other, but that it is also flipped from a left-to-right to a right-to-left orientation. As mentioned previously, run 520 is meant to be displayed in

left-to-right fashion. Figure 5C illustrates the final display of the text. Run 520 has been flipped again, but this time the flip is performed about the axis along the center of run 520, so that it stays in place relative to the other runs (510 and 530).

Turning now to Figure 6, an embodiment of a system which may support bi-

5   directional text in legacy applications is illustrated. Many applications are written in portable code 610, which may be used on multiple devices without modification. The portable code 610 contains references to device-specific code, with the expectation that those references will be useful to accomplish well-defined tasks on the devices on which the portable code 610 is executed. Each device may be expected to have a

10   native operating system (OS) 630, and it may also have glue code 620, i.e. code which may form the interface between the portable code 610 and the native OS 630. Note that glue code 620 may be included with portable code 610 in some products, it may be standard glue code which is expected to be present on some devices, or it may be some combination of the two. Typically, the division between the portable code 610 on

15   one side and the glue code 620 and native OS 630 on the other side may be represented as the device layer 640.

To implement the support of bi-directional text in legacy applications, additional code may be added, in the form of layer 650. Layer 650, represented in one embodiment as a routine 'DevlTextDraw' with parameters of a string and a position,

20   may be used to perform the flipping and selective flipping discussed previously. It will be appreciated that the exact location of the layer 650 may not be apparent, as it may form part of the glue code 620 or it may be written in a more device-independent form such that it would appear to belong on the portable side of the device layer 640.

However, it will similarly be understood that layer 650 accepts a request to draw text which the portable code is written to draw in left-to-right fashion. Layer 650 then transforms the text accompanying the request such that it can then be sent to the glue code 620 or native OS 630 in a format such that the native OS 630 and glue code 620

5 can process the text as though the text came from an application that has bi-directional text capabilities. In so doing, the code, hardware, or whatever else may embody the method of flipping and flipping selectively provide support in a bi-directional text environment for a legacy application suitable for unidirectional text.

It will be further appreciated that a layer such as layer 650 may be useful in

10 devices such as cellphones, handheld computers or appliances, pagers, and other devices which have limited memory and processor resources. As such, layer 650 may function better in its environment if it does not use a lot of memory and if it utilizes a simple and straightforward process. So, it may be preferable to avoid the use of complicated data structures. Additionally, it may be expected that the size of the

15 display will be small enough that techniques which would be viewed as brute-force and computationally expensive may be well suited to embodiment in layer 650.

Turning to Figure 7, one embodiment of a method of formatting text from a legacy application for bi-directional text display in a monospace font is illustrated. At block 710, the text to be displayed is received. This text may be received as part of a

20 line with a location at which it is to be displayed, a full line, or a set of characters that span multiple lines. To the extent that the text received at block 710 spans multiple lines, it is broken into separate lines for processing.

At block 720, all of the text that was received is flipped on a line-by-line basis. In one embodiment, this is done by exchanging the contents of character locations which are directly opposite each other relative to the vertical center axis of the display on which the characters will appear.

5   After each line has been flipped, at block 730, all runs of foreign characters are identified. These runs may include spaces or punctuation, but will not be identified as spanning lines. Then, at block 740, each of the runs of foreign characters is flipped in place, such that those runs have their original left-to-right orientation. At this point, the text may be passed on to be displayed, as all of the characters are now in the proper

10   position for display. As will be appreciated, this process may be repeated as much as necessary for each succeeding set of text which is to be displayed.

Turning to Figure 8, one embodiment of a method of formatting text from a legacy application for bi-directional text display in a proportionally spaced font is illustrated. At block 800, the process begins. At block 810, all text to be displayed is

15   received. This may include both a string of characters and a start position for display in one embodiment, and may further include style information such as point size or which font to use.

At block 820, the text is broken into lines. In the process, the possibility of a run of foreign (left-to-right) text spanning two lines is eliminated. This means that a string of

20   foreign text which spans two lines will be broken into two strings, one on each line, which will be processed separately. At block 830, the first line of text is received for further processing. At block 840, the runs of left-to-right and right-to-left text are identified or generated.

At block 850, the positions of the runs are flipped about the vertical axis of the screen, as was discussed with respect to Figures 5A, 5B and 5C. Thus, each run is identified as a string of characters and a location, and then the locations are flipped about the axis, with the orientation flipped as well. Next, at block 860, the runs are then

5 rendered in accordance with their right-to-left or left-to-right characteristics. Thus, the runs with left-to-right characteristics are effectively flipped back to their original orientation for rendering purposes.

At block 870, a determination is made as to whether any more lines of text need to be processed. If more lines remain, the next line is effectively received for

10 processing at block 880, and the process then moves to block 840 for generation of runs within that line. If no more lines remain, the process terminates at block 890.

Turning to Figure 9, a machine-readable medium embodying a method of supporting bi-directional text is illustrated. Note that the method may be embodied in a medium in the form of instructions that a processor may execute. Execution of the

15 instructions may be expected to cause the processor to perform the method of supporting bi-directional text. The medium may be magnetic media such as tape or disk, optical media, electronic media such as random access or read-only memory (including EPROM and EEPROM), a carrier wave, or any other machine-readable medium. Similarly, the method may be embodied in instructions, the instructions may

20 be downloaded or transmitted from a server to a client, and those instructions may be executed by the processor of the client to cause the processor to perform the method. Such a client may be a remote client coupled to the server through a wireline or wireless link, such as a wireless device for example.

In the embodiment illustrated, medium 900 includes multiple modules. Init 910 is an initialization routine which may prepare a buffer or otherwise prepare a display. Clear 920 is a routine which may be used to clear the display. Draw 930 is a routine which may be used to actually draw a character on the display. IsLtoR 950 is a routine

5 which indicates whether a character should be displayed in a left-to-right fashion. Similarly, isRtoL 960 is a routine which indicates whether a character should be displayed in a right-to-left fashion and isEither 970 is a routine which indicates whether a character may be displayed as either left-to-right or right-to-left. Characters which may be displayed in either fashion are typically blank spaces or punctuation, but other

10 characters may be included as appropriate.

InvertCharacters 940 is a routine which inverts all of the characters within a line without respect to whether the characters should be displayed in a right-to-left or left-to-right fashion. FindandInvertLtoR 980 is a routine which finds runs of characters in a line which should be displayed in left-to-right fashion and inverts the characters within the run. Thus, using InvertCharacters 940 in concert with FindandInvertLtoR 980 may

15 result in the transformations previously outlined. Drawtext 990 is a routine which may be used to automate the calls or interfaces to the other routines. Finally, DevITextDraw 995 is a routine which may have a device independent interface to legacy applications and may in turn call Drawtext 990.

20 Turning to Figure 10, an embodiment of a conventional general purpose computer is illustrated. CPU 1010 is coupled to processor bus 1015, which in turn is coupled to host bridge 1050. Host bridge 1050 is coupled to memory 1030 and to both I/O bridge 1070 and PCI bus 1025. PCI Bus 1025 is coupled to PCI Agents 1020. I/O

bridge 1070 is coupled to keyboard 1081, mouse 1083, modem 1085 and disk drive

1087. The apparatus and function of the embodiment of Figure 10 will be well

understood by those skilled in the art.

Turning to Figure 11, an embodiment of a network in which a wireless device

5   may be used is illustrated. Mobile device 1110 may retrieve hypermedia information

(such as HTML documents, Compact HTML (cHTML) documents, Extensible Markup

Language (XML) documents, HDML documents, or WML documents for example) from

one or more network server devices. Network server devices 1150 and 1160 represent

the myriad network server devices which may be accessible through either public or

10   private networks, such as the Internet, or a corporate LAN. Network servers 1150 and

1160 may, for example, be general purpose computers or other devices capable of

communicating over a network and having accessible information suitable for

transmission over a network.

Mobile device 1110 has a display 1115 and a user interface 1117. Additionally,

15   in one embodiment, mobile device 1110 may include a microbrowser, such as the

UP.Browser microbrowser offered by Phone.com of Redwood City, California. Such a

microbrowser may be stored in a local memory of the mobile device 1110, which

enables the mobile device 1110 to access and retrieve hypermedia information from

network servers such as network servers 1150 and 1160.

20   The communication path between mobile device 1110 and network servers 1150

and 1160, in the embodiment illustrated, includes airnet 1120, proxy server device

1130, and landnet 1140. It will be appreciated that other communications paths may be

utilized. Airnet 1120 is a wireless communications network, such as a cellular digital

packet data network (CDPD), a Global System for Mobile (GSM) network, a Code

Division Multiple Access (CDMA) network, or a Time Division Multiple Access (TDMA)

network for example. The communications protocols used by airnet 1120 may include,

for example, WAP and/or HDTP. Airnet 1120, in one embodiment, may be utilized to

5   transmit and receive voice or other data to and from wireless device 1110, along with

hypermedia data.

Proxy server device 1130 may be, for example, a conventional general purpose

computer, such as a workstation or PC. Proxy server device 1130 acts as a bridge

between airnet 1120 and landnet 1140. It will be appreciated that the same equipment

10   may be suitable for the roles of both the proxy server device 1130 and network servers

1150 and 1160. In one embodiment, proxy server device 1130 transforms data

received via landnet 1140 into a form suitable for transmission over airnet 1120, and

similarly transforms data received via airnet 1120 into a form suitable for transmission

over landnet 1140.

15   Landnet 1140, in one embodiment, is a primarily land-based network that may

include the Internet or World Wide Web. Landnet 1140 may also include an intranet, a

local area network, or other suitable collection of connected machines. Communication

via landnet 1140 may be expected to utilize a protocol such as Transmission Control

Protocol (TCP/IP), HTTP, or Secure HTTP (sHTTP).

20   It will be appreciated that in an alternate embodiment, a browser or other

software may be hosted on a machine such as proxy server 1130 and used to control

the display 1115 of mobile device 1110. Whether the display 1115 is controlled within

the mobile device 1110 or at a remote location, the method and apparatus for

supporting bi-directional text may be used to support display of bi-directional text on the

mobile device 1110.

Turning to Figure 12, a block diagram of one embodiment of a mobile device

such as mobile device 1110 is illustrated. The mobile device 1110 includes a processor

5   1210 which may be or include any of: a general- or special-purpose programmable

microprocessor, a Digital Signal Processor (DSP), an Application Specific Integrated

Circuit (ASIC), a Field Programmable Gate Array, a Complex Programmable Logic

Device, or any combination of these devices. Mobile device 1110 also includes a

wireless control protocol (WCP) interface 1260 which couples to a carrier network via

10   airnet 1120 to receive incoming and send outgoing signals. Device Identifier (ID)

storage 1270 stores and supplies to WCP interface 1260 a device ID suitable for use in

identifying the mobile device 1110 to outside entities.

Additionally, mobile device 1110 includes memory 1230, which stores data

and/or software, keypad interface 1220 which couples the processor 1210 to the user

15   interface 1117, and display 1290 which couples the processor 1210 to the display 1115.

Mobile device 1110 may also include encoder/decoder 1240 and voice circuitry 1250

which may be utilized to transmit and receive voice signals to and from the user of the

mobile device 1110. These voice signals may be either received from or transmitted to

the airnet 1120 as appropriate.

20   It will be appreciated that a microbrowser may be part of the software stored in

memory 1230, and that software suitable for implementing the method for supporting bi-

directional text may also be stored in memory 1230. Similarly, logic or software (such

- 17 -

as firmware for example) may be incorporated into the display interface 1290 to implement the method for supporting bi-directional text.

In the foregoing detailed description, the method and apparatus of the present invention has been described with reference to specific exemplary embodiments

5   thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the present invention. In particular, the separate blocks of the various block diagrams represent functional blocks of methods or apparatuses and are not necessarily indicative of physical or logical separations or of an order of operation inherent in the spirit and

10   scope of the present invention. For example, the various blocks of Figures 6 or 9 may be integrated into components, or may be subdivided into components. Moreover, the blocks of Figure 7 represent portions of a method which, in some embodiments, may be reordered or may be organized in parallel rather than in a linear or step-wise fashion. The present specification and figures are accordingly to be regarded as illustrative

15   rather than restrictive.